

a FIELD OF THE INVENTION PROGRAMMING DEVICE

a ^{Present}
The ¹invention relates to a programming device having a software tool for processing objects, of which a first object, equipped with data, is provided as model for a second object that has a pointer with which, at the beginning of an access to the second object, the data of the first object can be incorporated into the second object by the software tool.

a BACKGROUND OF THE INVENTION

a Data are often stored in a first object in the form of a type and in a second object in the form of an instance, the type being provided as model for the instance. An instance usually comprises both the instance-specific and the type-specific data, the result being that the type-specific data need to be stored several times in a memory of the programming device. If, for example, n instances are applied to a type, the type-specific data usually need to be stored (n+1) times, which means a very large data volume.

a SUMMARY

It is ^{an} ~~the~~ object of the ^{Present} ~~1~~ invention to reduce the data volume in a programming device when creating multiple instances from a type.

a This object is achieved with a programming device ^{according to the} ~~of the kind~~ ¹ ~~present invention~~ ¹ ~~as initially~~.

a ^{Sub} ~~Only~~ the instance-specific data and a reference (pointer) to the type are deposited in the instance. The type-specific data are not incorporated into the instance until a data access to the instance begins.

a In an embodiment of the ^{Present} ~~1~~ invention, provision is made for at least one data group of the first object to be depositable by the software tool into the second object, the result being that at the beginning of an access to the second object, the software tool does not incorporate this data group of the first object into the second object.

This makes it possible for a data modification in a type to have either an influence or no influence on the data in an instance constituted with that type. A programmer can thus choose whether future data modifications in the type are to have an affect on the instance data.

BRIEF DESCRIPTION OF THE DRAWINGS

~~The invention, its embodiments, and its advantages will be explained in more detail below with reference to the drawings, in which an exemplary embodiment of the invention is illustrated, and in which:~~

Figures 1 and 2 show, in a schematic depiction, a memory of a programming device having software objects.

DETAILED DESCRIPTION

In Figure 1, the reference number 1 designates a main memory of a programming device into which a software tool 2, a first object in the form of a type 3, and a second object in the form of an instance 4 are loaded. Type 3 serves as model for instance 4, and comprises a type identifier Tk and data subdivided into data groups 6b through 6f, a data group comprising, for example, a width of 8 bits or a width of only one bit. Prior to a data access, instance 4 has only instance-specific data 7 and an attribute list, which in the present example comprises six fields 8a, 8b through 8f. Field 8a is occupied by a pointer Id for addressing type 3 with type identifier Tk; fields 8b through 8f, which are associated with data groups 6b through 6f of type 3, are occupied by attributes which indicate whether, at the beginning of an access to the instance data, the data to be incorporated by software tool 2 in the event of a data modification in data groups 6b through 6f of type 3 are the modified data or the data valid prior to that modification.

It is assumed hereinafter that fields 8b, 8c, and 8d are occupied by an attribute 1, and fields 8e and 8f by an attribute 0. This means that the data modifications in data groups 6b through 6d of type 3, but not modifications in data

a groups 6e and 6f, are to be taken into account upon incorporation into instance 4. ^{AS}~~as~~ as a result, it is necessary for software tool 2 to write data groups 6e and 6f into instance 4 prior to any modification in those data. It is also
5 assumed that the software tool created data groups 6b through 6f at a time t0 and modified them at a time t1, thereby deleting the data valid at that time. The modified data groups are labeled 6b' through 6f'. Before a user can access the instance data, the type data must first be incorporated by
10 software tool 2 into instance 4 (Figure 2). Corresponding to the occupancy of fields 8b through 8f of the attribute list, software tool 2 incorporates into instance 4 only the type data valid at time t1 (i.e. data groups 6b', 6c', and 6d'), but not the type data 6e' and 6f' valid at that time t1. After
15 incorporation of the type data, the instance data which a user can now access comprise both the instance-specific data 7 and the type-specific data at time t0 and time t1, i.e. data groups 6b', 6c', 6d', 6e, and 6f.

a 20 The ^{present} invention considerably reduces the quantity of data that must be saved, since only the data modifiable at an instance are stored. The type data are incorporated into the instance only prior to an access to the instance data. A data modification to a type can selectably affect the instance
25 data.